INTERNET-BASED INTERACTIVE AUDITORY VIRTUAL ENVIRONMENT GENERATORS

Christian Borß, Andreas Silzle, and Rainer Martin †*

Institute of Communication Acoustics
Ruhr-Universität Bochum, 44780 Bochum, Germany
{firstname}.{lastname}@rub.de

ABSTRACT

In this paper we investigate general design considerations and practical implementation aspects for Internet-based interactive auditory virtual environments (I-AVE) for the post-PC era. An implementation of such an AVE generator as a web service allows for platform independent "AVE services" for mobile devices almost "anywhere on any device" using a standard web browser. We propose a client-server architecture which computes the acoustic signals on a high-performance server and provides low-latency audio streaming from the server to the client.

1. INTRODUCTION

Auditory virtual environments (AVE) are used in a variety of applications, for example, in architectural acoustics, in entertainment and 'edutainment' systems. The purpose of such systems is to generate sound events which are perceived as being close to the auditory percept in natural environments [1, 2]. The sound can either be reproduced by loudspeakers or by headphones. For the latter case or for loudspeaker reproduction with crosstalk cancellation, head-related signals are required. For headphone reproduction, the generation of these binaural signals must also include the simulation of sound propagation in enclosed environments to incorporate reverberation which is present in physical environments.

The generation of an AVE is computationally demanding and cannot be easily implemented on end-user devices with limited computational power. On the other hand, it is desirable to make interactive AVEs accessible for low-end PCs, PDAs, Smart Phones etc. and thus extend the possible use cases. An interactive guide through virtual exhibition halls or edutainment software for mobile devices are examples for such an application. While the purpose of the conventional World Wide Web was to serve and link documents, the second generation World Wide Web ("Web 2.0") is orientated towards providing services and linking people. An implementation of an AVE generator as web service could allow for platform independent "AVE services" for these use cases.

For some of the above mentioned applications off-line computation of the binaural signals and subsequent playback suffices. However, the user's Quality of Experience, a term introduced in [3], can greatly be enhanced by interactivity which requires online computations. E.g., if the listener can change the virtual head orientation, the front-back confusion can be reduced which is a well known problem when using non-individual head-related impulse responses (HRIRs) [4, 1]. In an interactive AVE the position of the sound source, the position of the receiver, or the acoustic properties of the enclosure may be changed at any time by the user. For consistent interaction, the acoustic signals which are offered to the listener must reflect these changes immediately. Thus, the responsiveness of an interactive AVE has a significant impact on the user's Quality of Experience. Any client-server implementation must therefore be optimized for low latency.

2. CONVENTIONAL AVE GENERATORS

The AVE generators which are considered in this work comprise one or several sound sources, a single human receiver, and reflective boundaries. Under the constraints of a geometric propagation model the reflections may be computed using the mirror image model or the ray-tracing technique [5, 6]. The simulation of the early reverberation thus takes the directivity of the source, the reflections at the boundaries, and the sound attenuation and propagation delays into account. The directional characteristics of human hearing are modelled by HRIRs which characterize the sound transmission from a source in the environment to a reference point in the ear canals under anechoic conditions [1, 7]. For headphone reproduction, these impulse responses are necessary for an out-of-head localization. Off-line and real-time AVEs of this kind have been developed and described thoroughly [8, 9, 10, 11].

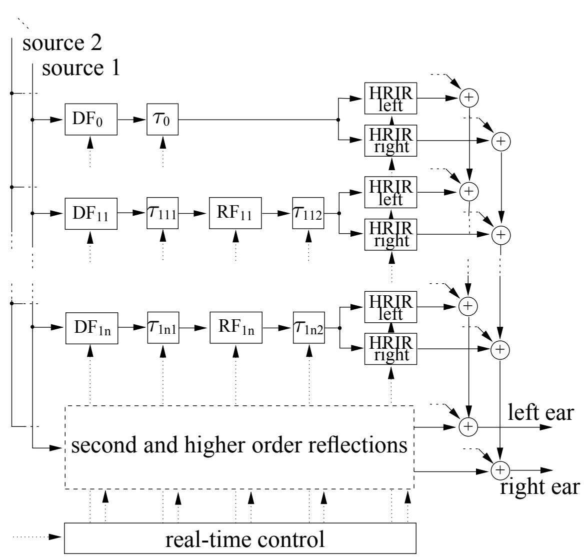

Figure 1: Signal processing model of the AVE generator IKA-SIM for the early reverberation

*Now with Fraunhofer IIS Laboratories, Am Wolfsmantel 33, 91058 Erlangen, Germany, eMail: andreas.silzle@iis.fraunhofer.de

† This work was funded in part by grants from the European Union FP6, Project 004171 HEARCOM (http://www.hearcom.eu/).

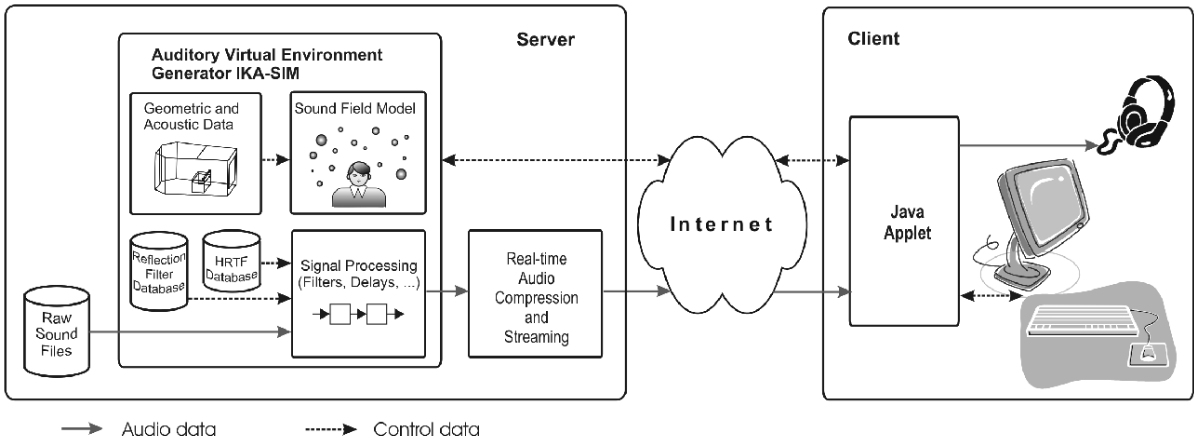

Figure 2: Structure of an interactive AVE generator over the Internet

The cited systems share the underlying physical model but differ in the signal processing structure and the strategy how the propagation paths are determined. Beam-tracing and binary space partitioning can be used to reduce the required computational complexity [12]. These precalculation techniques are especially beneficial for distributed AVEs [13, 14]. The reduced computational complexity for complex geometries is one of the major advantages of the ray/beam-tracing technique [15]. An advantage of an interactive AVE based on a mirror image model is the possibility to easily take the Doppler shift into account for a moving receiver or for moving sound sources. Although the Doppler shift may be below the perception threshold for slow movements, an asynchronous resampling of the mirror image signals is necessary to avoid processing artifacts. This can be implemented quite efficiently due to the parallel structure of the mirror image signal processing model [16].

Figure 1 details a block diagram of the AVE generator IKA-SIM [17] for the early reverberation. It exemplifies the signal processing for a mirror image model. The source directivity and the boundary reflections are modeled by filters DFx and RFx, respectively, where x denotes the order of the reflection (only the direct path and the first order reflections are explicitly shown). Propagation delays are denoted by τ_x. In order to allow fractional delays, these are also realized by means of FIR filters. Note that the number of simultaneously active sources and the number of parallel branches, each of which simulates an acoustic path from the source to the ear canal, is limited by the computational power available for the AVE generator. For an interactive AVE all parameters such as delay, directivity, and reflection filters as well as HRIRs must be adapted when the source or the receiver position is changed by the user. The parametric reverberator which models the late reverberation does not depend on user interaction and thus does not have to be adapted if the late reverberation is ideally diffuse.

3. SYSTEM CONCEPTS FOR INTERNET-BASED INTERACTIVE AVE GENERATORS

To provide AVEs for end user devices with limited computational power, the computational demanding part could be taken over by a separate compute server. For the design of such a system several objectives must be balanced such as

- the computational load on the client (and the server)
- the latency (closely related to usability issues)
- the bit rate on the link between client and server
- the user's Quality of Experience for the given application.

The client-server approach requires the distribution of all computations such that the above objectives which are discussed in greater detail in the following sections are fulfilled. With respect to a low computational power of the client and a low delay between user action and auditory perception two concepts seem to be feasible:

A) The binaural signals are completely computed on the server including direct path, early and late reflections, and HRIRs and are streamed to the client.

B) The signals corresponding to the direct path including the HRIRS are computed on the client, the signal components corresponding to early and late reflections and their HRIRS are computed on the server and are streamed to the client.

Concept A which is illustrated in Figure 2 provides for a conceptually clear separation between the generation of the auditory signals and the user interface of the client. However, the overall latency and thus the usability of the distributed AVE largely depends on the latency of the audio streaming which cannot be made arbitrarily small. Furthermore, a general purpose audio coder must be employed to support different types of audio sources.

Concept B allows for immediate feedback to user interactions. The acoustic signal on the direct path provides the most prominent localization cues with almost no delay. Lagging early and late reflections may be tolerated to some extent. This solution does also not require extensive computational power on the client. However, it requires the transfer and storage of the HRIR database on the client and possibly a higher bit rate on the link between server and client for more than one sound source.

3.1. Computational Complexity of AVE Generators

Despite several simplifications and powerful computers, the implementation of an interactive AVE generator is still a computational demanding task. The operator of an AVE service has to find a compromise between the computational complexity for each single AVE and the number of clients who can concurrently use the service. The rendering of the mirror image sources is the computationally most expensive part of a real-time AVE which is implemented as described in section 2. The computation of each mirror

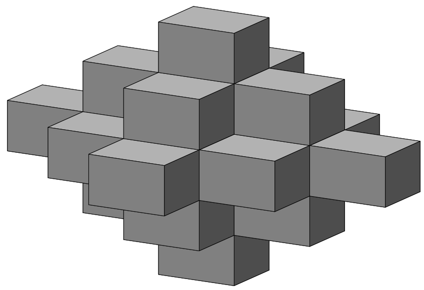

Figure 3: Geometrical structure which results from all mirror image rooms with an order of less than or equal to two for a rectangular room. The computational complexity of the mirror image model is proportional to the number of mirror image rooms.

image source requires a directivity filter, reflection filters, time interpolation filters, and the HRIR for the left and the right ear. For a simple rectangular room with six surfaces the number of mirror sources $M(o)$ up to order o can be directly calculated,

$$M(o) = 2o^2 + 2o + 1 + 2\sum_{i=0}^{o-1}\left(2i^2 + 2i + 1\right) \quad (1)$$

$$= \frac{4}{3}o^3 + 2o^2 + \frac{8}{3}o + 1 \quad . \quad (2)$$

The corresponding mirror image rooms form a double pyramid structure as shown in Figure 3 for $o = 2$. We have derived equation 1 from this geometrical structure by adding the number of mirror image rooms in each layer. Based on the consideration that a limited sum over a second order polynomial yields a third order polynomial, we attained the simplified form 2. The polynomial coefficients were determined from 4 values obtained from equation 1 ($M(0) = 1$, $M(1) = 7$, $M(2) = 25$, $M(3) = 63$).

For an implementation with uniform filter length this yields an

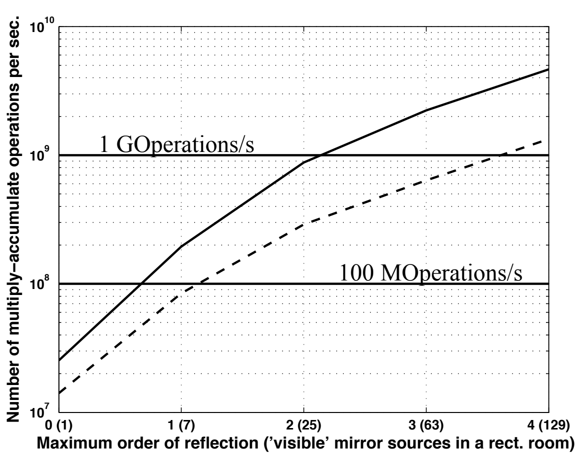

Figure 4: Estimated number of MAC-operations for the signal processing part of a real-time AVE generator ($f_s = 44.1$kHz) for uniform and non-uniform HRIR filter lengths as published in [18]. Solid: Implementation with uniform filter lengths. Dashed: Implementation with non-uniform filter lengths.

algorithmic complexity of $\mathcal{O}(N^3)$ for mirror sources up to order N. Figure 4 shows an estimation of the computational load for an implementation with uniform and non-uniform filter lengths as published in [18]. The author assumes that besides the reflection filters all filters are implemented as FIR filters and one operation is defined as a multiply-accumulate operation (MAC). We observe that a typical mobile device with, e.g., a 300MHz RISC processor that cannot perform single cycle floating point MAC operations would hardly be able to render second or higher order reflections. On the other hand, the signal processing for one propagation path, i.e. the direct path in concept B, would not increase the computational load significantly.

For more complex or non-rectangular rooms with n_s surfaces there are a maximum of ${n_s}^o$ order o mirror sources. In this case a visibility check has to be performed for each mirror image source resulting in an exponential algorithmic complexity [19]. For the simulation of high order reflections in complex geometries this visibility check thus has a major impact on the computational complexity for a mirror image model [15]. A ray-tracing algorithm does not require such a visibility check and thus is better suited for this case. Furthermore, in complex geometries mirror image sources can suddenly become visible or invisible when the sound source or the listener moves. If diffraction effects are not taken into account, this may result in audible artifacts or a degradation of plausibility. A simulation of diffraction effects is feasible but also increases the computational complexity [20, 21].

By restricting the geometry to a rectangular room, these problems are avoided. In addition to the omission of the visibility check, the determination of the mirror image source positions has a lower computational complexity. These savings result in a larger number of clients who can use the discussed AVE service concurrently.

We conclude that the computational complexity mainly affects the server and that a restriction to a rectangular room can enlarge the number of concurrently served clients if such a restriction is admissible. The simulation of one single reflection as performed in concept B for the direct path signal has a relative low computational complexity. Depending on the computational capability of the client, this has only a limited impact on the client load. Therefore, with regard to computational load on the client, both concepts are similar.

3.2. Network Transmission Delay

For an interactive AVE the responsiveness is an important factor for the perceived quality [22]. The responsiveness describes the time delay between an action or an event and the corresponding change of the output as perceived by the listener. The threshold for latency perception for a system with head rotation has been determined to be about 90ms [23]. The latency of a client-server based system will be clearly above this threshold. It consists of

1. the command transmission to the server,
2. the audio rendering on the server,
3. the audio coding,
4. the transmission over the network,
5. the jitter buffer,
6. the audio decoding, and
7. the playback buffer of the client.

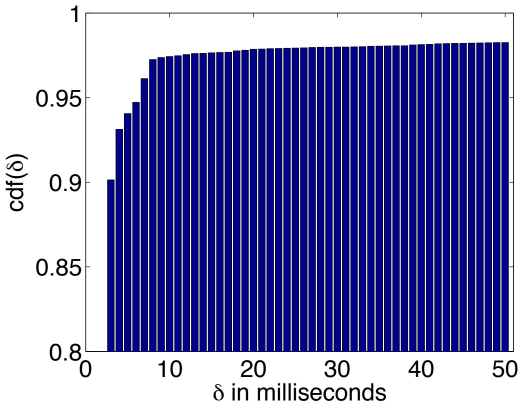

Figure 5: Cumulative distribution of the network transmission jitter δ for European DSL and cable connections (determined from 43 test series)

In contrast to the coding and decoding delay which can be reduced by means of a low-delay audio coder [24], the delay caused by the transmission over the network cannot easily be reduced. We have measured the delay of packets sent over the Internet and its variation to investigate the requirements for the playback of the audio stream. Assuming that the packet delay does not depend on the transmission direction, the average packet delay $\overline{t}$ can be estimated by the measurable average roundtrip time $\overline{t_r}$, as $\overline{t} \approx \frac{\overline{t_r}}{2}$. The delay t_i of packet i is composed of a constant network latency T and a variable part δ_i which depends on the load of the routers and as such can be modeled as a stochastic process, $t_i = T + \delta_i$.

The buffer time T_B which is needed by the receiver to compensate the delay jitter, should be adapted to the stochastic properties of the variable delay δ_i. Figure 5 shows the cumulative distribution of the variable delay δ_i we have determined for selected DSL and cable modem lines in Europe. It discloses the percentage of packets which are not received on time for a given buffer time. The result is determined from 43 test series (Germany: 30, Denmark: 8, United Kingdom: 4, Netherlands: 1) of 200 measured values each. The roundtrip time was measured by a host at the Ruhr-Universität Bochum which is connected to the German national research and education network DFN. As a transparent encoding of the binaural signals demands a rather high bit rate, only lines with a channel capacity of at least 192 kbit/s were investigated.

The stochastic property of the network transmission delay can be used to tune the jitter buffer of the receiver for latency reduction. 40 of the 43 test series had a maximum variable packet delay $\max(\delta_i)$ of less than 21ms. Therefore the jitter buffer of the receiver can be reduced to a value that leads to an acceptable overall system latency for both concepts. Concept B is less dependent on transmission latencies. However, both concepts have the same latency with regard to changes to the sound source itself. Only the *position* of the 0th order mirror source (direct path) can be changed immediately in concept B.

3.3. Bit Rate

To support different types of audio sources, a general purpose audio coder must be employed for concept A. Furthermore, a transparent encoding of the binaural signals generally demands a rather high bit rate. The required bit rate for concept B is higher than the one for concept A and depends on the number of sound sources in the AVE as an audio stream is required for each source. Some of this overhead may be reduced, if the coders for each of these streams are adapted to the type of source signals. For example a speech coder might be used if a sound source is known to comprise clean speech.

3.4. User's Quality of Experience

Similar to many Internet-based real-time services such as Voice over IP, the system latency is of critical importance. To allow the user to perform a specific task in the AVE the responsiveness of the AVE must be ensured. The admissible latency is highly dependent on the application and it's scope. E.g., an Internet-based localization adjustment test which is based on an interactive AVE can still be conducted at latencies of up to 1500ms without significant deviations of the test results [25]. However, such high latencies reduce the user's Quality of Experience. To avoid user annoyance, the overall delay between user interaction and audio output should be less than 600ms for the given use case [25, 26]. Most streaming solutions, which were developed for non-interactive use like internet-radio, have an overall latency of more than 1s and are therefore not suitable for such an application.

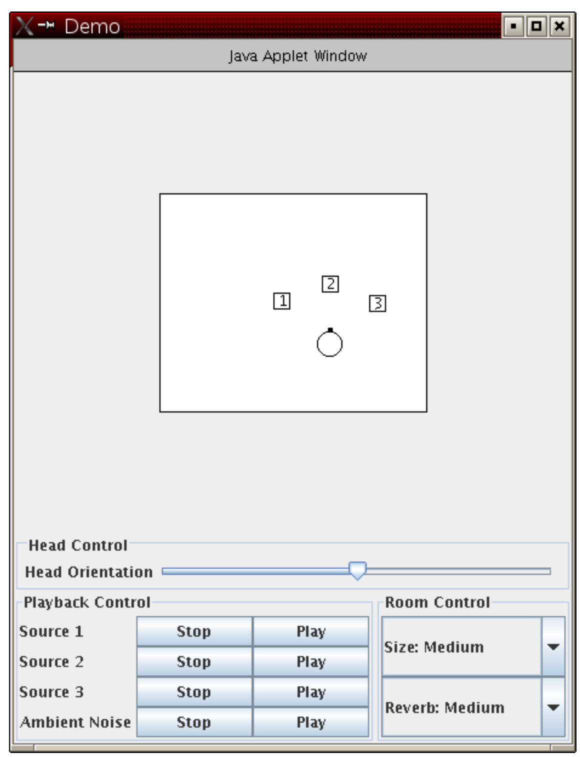

Figure 6: Example application for an "AVE service" which can be used to demonstrate issues of ambient acoustics

4. IMPLEMENTED SYSTEM

We have implemented a platform independent “AVE service” using concept A. The implemented system can be used to demonstrate issues of ambient acoustics without high computational requirements or prior software installation on a client. We have developed two web applications which use our “AVE service” for different purposes - a self-screening localization test described in the following section and a generic demonstrator.

The user interface of the interactive demonstrator is shown in Figure 6. The horizontal projection of the room is displayed in the upper part of the applet. The listener is symbolized by a circle and a black dot which denotes the head orientation. Three sound sources are symbolized by numerated boxes. The position of the listener as well as the sound sources can be changed by “drag and drop”. The lower part of the applet contains further control elements. The head orientation can be changed with a horizontal slider. Each sound source can be activated or deactivated with the corresponding “Play” and “Stop” buttons. In addition to the three movable sound sources, ambient noise can be added using preprocessed sound files. The room size which affects the generation of the early reflections can be changed with the “Size” drop-down list. It contains presets for a small, a medium, and a large room. The late reverberation generator can be influenced by the “Reverb” drop-down list which contains presets for low, medium, and high reverberation portion.

4.1. Internet-Based Self-Screening Test

An AVE generator can be used to simulate room acoustics and to demonstrate communication conditions of various difficulties in daily life situations. As communication in adverse conditions is especially difficult for hearing impaired persons, an AVE could also be used for self-screening hearing tests. Such situations occur when several persons are talking at the same time (e.g., cocktail party or cafeteria), in the presence of loud background noise (e.g., traffic in the street), or in environments with strong reverberation (like in large rooms, train stations, etc). One of the aims of the HearCom project [27], for which the described client-server architecture is developed, is to provide such self-screening tests over the Internet and to raise awareness on difficult hearing situations. If in such a self-screening hearing test the answers of a test person deviate from the answers of a normal hearing reference group, a hearing impairment is indicated.

Figure 8: Screenshot of the self-screening localization test using an interactive AVE over the Internet

Figure 7: Overview of the virtual livingroom

Figure 7 shows the livingroom we have chosen as daily life environment in which ringing telephones are used for a virtual localization adjustment test. The listener is placed in the middle of the virtual room and has the possibility to turn his/her head.

Figure 8 shows the graphical user interface (GUI) we developed to visualize the scene. The user can interact with the environment by changing the virtual head orientation with the slider. The visualization of the scene is updated according to the head orientation - both, auditory and visual, virtual environments are synchronized.

We have chosen ringing telephones as stimuli and background music as interferer for this novel virtual localization test. All sound sources are simulated by the AVE generator IKA-SIM [17], which calculates the mirror image sources and the sound propagation. The late reverberation tail is generated by a parametric reverberation model. The output of this interactive environment is played back over headphones.

4.2. Implementation Details

In our implementation the client controls the AVE generator IKA-SIM on the server via XML structured commands. The binaural signals computed on the server are presented to the user in a Java environment. By using Java, system independence on the client side is achieved, and, as all necessary libraries for the execution are provided by the web-server, no additional software has to be installed by the user. This simplifies the use of such a web-service for a target group that does not have much experience in software installation.

The audio stream is encoded at 192kbit/s average bitrate using an open-source MP3 codec, *LAME* [28]. The MP3 frames are transmitted directly to the client using the realtime streaming protocol (RTP) [29] and the user datagram protocol (UDP). This combination is with regard to latency superior to systems using the transport control protocol (TCP), as no packets are retransmitted in the case of packet loss.

Because the Java 2 standard edition used by the web browser does not support the *real-time streaming protocol* (RTSP) [30] and the *RTP control protocol* (RTCP) [29], we have developed a simple replacement for these protocols which are used to initialize the audio streaming. The support for these protocols as well as the MP3 decoder are provided by libraries written in pure Java and are loaded with the Java applet from the web server. For the decoding of the MP3 frames a modified version of the open-source MP3 decoder JLayer [31] is used.

4.3. System Latency

A measurement of the system latency for a client and server connected to the Internet cannot easily be done if these are not at the same site. However, the latency can be measured, if the server and client are close together and directly interconnected by a local area network (LAN). We have measured the delay as shown in Figure 9 using a periodic impulse that is simultaneously streamed to the client over the network and output over the parallel port of the server. The delay for output via parallel port can be neglected as it is around 1μs. A 100BASE-TX Ethernet switch has a network latency of less than 1ms for a light load. By this, the factors that contribute to the total delay can be narrowed down to the rendering, encoding, buffering, decoding, and playback of the audio output. The server has to render an audio frame in less than one frame period, i.e. less than 26ms for MP3 frames. By deactivating the bit reservoir of the MP3 encoder, the latency caused by the encoder can be reduced to 26ms which is equivalent to the decoder latency. The internal buffers of the Java sound implementation, the buffer of the underlying system dependent interface (e.g. DirectSound), and the driver's buffer cannot be reduced and contribute about 100ms-150ms of additional delay for the Windows version of Java 1.5.0-06. For Java applications this part can be further reduced with the Java Native Interface (JNI) and a system dependent sound API but not for Java applets. The execution of native code would compromise the security restrictions of the webbrowser, because it is not executed within the Java sandbox. The jitter buffer, which contains a whole number of audio frames to deal with the varying delay of the transmitted packets, contributes at least 52ms. This delay is caused by a minimum of two MP3 frames which are necessary even for a very small variation of the

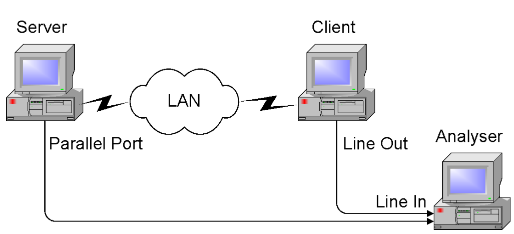

Figure 9: Setup for measuring the total latency of the implemented system using a periodic signal that is simultaneously output via the parallel port of the server

	latency contribution in ms
command transmission to the server	~ 40
audio rendering	< 26
audio coding	26
transmission over the Internet	~ 40
jitter buffer	≥ 52
audio decoding	26
playback (Java Sound)	100-150 (typical)

Table 1: Contribution to the total latency of the implemented system using an MP3 codec with a block length of 26ms and a DSL connection

transmission delay to avoid a buffer underrun if the frames are played back immediately after reception.

We have measured a latency of 350ms excluding the transmission delay for a client using a Java 1.5.0-06 runtime environment from Sun Microsystems, Inc., Microsoft WindowsXP SP2, an AC97 onboard sound card, and a jitter buffer suitable for DSL lines. Table 1 gives an overview of the contribution to the total latency of the various elements of the implemented system. For a typical DSL line the latency requirements with respect to the user's Quality of Experience as found in [25] are fulfilled.

5. CONCLUSIONS

The implementation of an AVE generator as web service makes interactive AVEs accessible for low-end PCs and mobile devices. Two implementation concepts (the binaural signals are completely computed on the server (A) vs. the direct path is computed on the client and the remaining part on the server (B)) for this novel approach were discussed. Although concept B has great appeal in terms of a low-latency direct acoustic path, the bit rate overhead, the necessity to transfer the HRTF database to the client, and practical implementation aspects (like mixing different audio streams with different coding latencies) make concept A more viable. For an application with limited latency requirements we propose a system based on concept A which results in a good balance between latency, quality, and computational complexity.

Two applications based on concept A were presented to demonstrate the potential and the benefit of an interactive "AVE service" which can be used almost "anywhere" on "any device" with a standard web browser.

We have determined the latency of the implemented system which is of critical importance for an interactive AVE. We measured a latency of 350ms in a local area network using a jitterbuffer suitable for DSL lines. We also measured the network transmission delay for DSL and cable modem lines in Europe. From both measurements we can conclude that the resulting latency is admissible for the intended use-cases.

6. REFERENCES

[1] J. Blauert, *Spatial Hearing, The Psychophysics of Human Sound Localization*. Cambridge, MA: MIT Press, second ed., 1997.

[2] P. Novo, "Auditory Virtual Environments," in *Communication Acoustics* (J. Blauert, ed.), Berlin-Heidelberg-New York: Springer Verlag, 2005.

[3] ITU-T (Nortel Networks), *COM 12-D95 - Quality of Experience*. Intern. Telecom Union, Geneva, 2003.

[4] G. Boerger, P. Laws, and J. Blauert, "Stereophone Kopfhörerwiedergabe mit Steuerung bestimmter Übertragungsfaktoren durch Kofpdrehbewegungen

(Stereophonic Headphone Reproduction with Control of Certain Transfer Factors by Head Rotations)," *Acustica*, vol. 39, pp. 21–26, 1977.

[5] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[6] A. Krokstad and S. Strom, and S. Sorsdal, "Calculating the Acoustical Room Response by the Use of a Ray Tracing Technique," *J. Sound Vib.*, vol. 8, no. 1, pp. 118–125, 1968.

[7] Møller, H. and Sørensen, M. F. and Hammershøi, D. and Jensen, C. B, "Head-Related Transfer Functions of Human Subjects," *J. Audio Eng. Soc.*, vol. 43, pp. 300–321, May 1995.

[8] J. Blauert et al., "Binaural Room Simulation," *Acustica - acta acustica*, vol. 72, pp. 295–296, 1991.

[9] G. M. Naylor, "ODEON - another hybrid room acoustical model," *Applied Acoustics*, vol. 38, no. 2-4, pp. 131–143, 1993.

[10] Vorländer, M., "International round Robin on room acoustical computer simulations," in *Proc. 15th International Congress on Acoustics (ICA)*, (Trondheim, Norway), pp. 689–692, 1995.

[11] L. Savioja, *Modeling Techniques for Virtual Acoustics*. PhD thesis, Helsinki University of Technology, Helsinki, Finnland, 1999.

[12] D. Schröder and T. Lentz, "Real-Time Processing of Image Sources Using Binary Space Partitioning," *Journal of the Audio Engineering Society*, vol. 54, no. 7/8, pp. 604–619, 2006.

[13] T. A. Funkhouser and P. Min and I. Carlbom, "Real-Time Acoustic Modeling for Distributed Virtual Environments," in *Siggraph 1999, Computer Graphics Proceedings*, (Los Angeles, CF, USA), 1999.

[14] R. Kajastila at al., "A Distributed Real-Time Virtual Acoustic Rendering System for Dynamic Geometries," in *122nd AES Convention*, (Vienna, Austria), May 2007.

[15] H. Lehnert, "Real-Time Generation of Interactive Virtual Auditory Environments," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY, USA), Oct. 1993.

[16] C. Borß, "Effcient Asynchronous Resampling of Audio Signals for Spatial Rendering of Mirror Image Sources," in *Proc. Annual Meeting of the German Acoust. Society (DAGA)*, (Dresden, Germany), Mar. 2008. URL: `http://www2.ika.rub.de/publications/2008/borss08efficient.pdf`.

[17] A. Silzle, P. Novo, and H. Strauss, "IKA-SIM: A System to Generate Auditory Virtual Environments," in *116th AES Convention*, (Berlin, Germany), 2004.

[18] A. Silzle, *Generation of Quality Taxonomies for Auditory Virtual Environments by Means of Systematic Expert Survey*. PhD thesis, Institut für Kommunikationsakustik, Ruhr-Universität Bochum, Bochum, 2007.

[19] J. Borish, "Extension of the Image Model to Arbitrary Polyhedra," *J. Acoust. Soc. Am.*, vol. 75, pp. 1827–1836, Mar. 1984.

[20] U. P. Svensson and R. I. Fred, "An Analytic Secondary Source Model of Edge Diffraction Impulse Responses," *J. Acoust. Soc. Am.*, vol. 106, no. 5, pp. 2331–2344, 1999.

[21] V. Pulkki, U. P. Svensson, and T. Paatero, "Efficient Representation of Edge Diffraction Impulse Responses," in *19th International Congress on Acoustics*, (Madrid, Spain), Sept. 2007.

[22] T. Djelani, C. Pörschmann, J. Sahrhage, and J. Blauert, "An Interactive Virtual-Environment Generator for Psychoacoustic Research II: Collection of Head-Related Impulse Responses and Evaluation of Auditory Localization," *Acustica - acta acustica*, vol. 86, pp. 1046–1053, 2000.

[23] P. Mackensen, *Auditive Localization. Head Movements, an Additional Cue in Localization*. PhD thesis, Fakultät I - Geisteswissenschaften, Technische Unversität, Berlin, 2004.

[24] M. Lutzky, "Fraunhofer IIS ULD Audio Software. Ultra Low Delay (ULD) Audio Encoders and Decoders," tech. rep., Fraunhofer Institut für Integrierte Schaltungen IIS, 2004. URL: `http://www.iis.fraunhofer.de/amm/download/paper/dl.html?f=wp_iis_uld_1.5b.pdf`.

[25] A. Silzle and C. Borß, "Latency Annoyance of Interactive Auditory Virtual Environments," in *Proc. Annual Meeting of the German Acoust. Society (DAGA)*, (Stuttgart, Germany), Mar. 2007. URL: `http://www2.ika.rub.de/publications/2007/silzle07latency.pdf`.

[26] Y. Suzuki and S. Yairi and Y. Iwaya, "Effect of Large System Latency of Virtual Auditory Display on Listener's Head Movement in Sound Localization Task," in *13th Int. Conf. on Auditory Display*, (Montreal, Canada), June 2007.

[27] *The HearCom Project Home Page*. URL: `http://www.hearcom.eu/`.

[28] *The LAME Project Home Page*. URL: `http://lame.sourceforge.net/`.

[29] *RTP: A Transport Protocol for Real-Time Applications*, Jul. 2003. RFC 3550, URL: `http://www.rfc-editor.org/rfc/rfc3550.txt`.

[30] *Real Time Streaming Protocol (RTSP)*, Apr. 1998. RFC 2326, URL: `http://www.rfc-editor.org/rfc/rfc2326.txt`.

[31] *The JLayer Project Home Page*. URL: `http://www.javazoom.net/javalayer/javalayer.html`.